A Hierarchy of Expressiveness in Concurrent Interaction Nets

Andrei Dorman[1] and Damiano Mazza[2]

[1] Dipartimento di Filosofia, Università degli Studi Roma Tre
and Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS
Andrei.Dorman@lipn.univ-paris13.fr
[2] CNRS, UMR 7030, LIPN, Université Paris 13, Sorbonne Paris Cité
Damiano.Mazza@lipn.univ-paris13.fr

Abstract. We give separation results, in terms of expressiveness, concerning all the concurrent extensions of interaction nets defined so far in the literature: we prove that multirule interaction nets (of which Ehrhard and Regnier's differential interaction nets are a special case) are strictly less expressive than multiwire interaction nets (which include Beffara and Maurel's concurrent nets and Honda and Laurent's version of polarized proof nets); these, in turn, are strictly less expressive than multiport interaction nets (independently introduced by Alexiev and the second author), although in a milder way. These results are achieved by providing a notion of barbed bisimilarity for interaction nets which is general enough to adapt to all systems but is still concrete enough to allow (hopefully) convincing separation results. This is itself a contribution of the paper.

Keywords: Interaction nets, Expressiveness in concurrency, Behavioral equivalences

1 Introduction

Interaction nets were introduced by Yves Lafont [10] as a model of distributed and deterministic computation, inspired by proof nets of multiplicative linear logic [11]. To this date, they have earned a prominent place in the theory of the optimal implementation of the λ-calculus (as *sharing graphs* [13, 2]) and functional programming languages in general [15, 16], as well as in the related field known as the geometry of interaction [7], for which they are the most natural syntax.

The main interest of interaction nets lies in the fact that they provide a simple yet extremely powerful paradigm for representing a vast variety of computational models, ranging from Turing machines to functional languages, passing through cellular automata and term-graphs, all this by respecting the essential idea that computation is local and the cost of elementary steps is bounded by a constant. Additionally, interaction nets provide a pleasant and intuitive graphical representation of programs, similar in style and spirit to string diagrams for monoidal categories.

Let us define interaction nets in a nutshell. We start with an *alphabet*, *i.e.*, a set of *symbols*, $\alpha, \beta, \ldots$, each with a given arity. Given a denumerable set of *ports* $x, y, z, \ldots$, the atomic components of interaction nets are:

- *agents* (or *cells*), of the form $\alpha(x; \tilde{y})$, where $x, \tilde{y}$ are ports, with the length of the list $\tilde{y}$ matching the arity of α, and x being the *principal port* of the cell;
- *wires*, which are multisets of exactly two ports, written $[x, y]$.

A *net* is a multiset of agents and wires, in which every port appears *at most twice*. We write nets as

$$\alpha_1(x_1; \tilde{y}_1) \mid \cdots \mid \alpha_m(x_m; \tilde{y}_m) \mid [z_1, u_1] \mid \cdots \mid [z_n, u_n],$$

using a notation reminiscent of process calculi (especially the solos calculus [14]). A port appearing exactly once in a net μ is *free*, and $\mathrm{fp}(\mu)$ is the set of free ports of μ; all other ports are *bound*, and may be renamed as usual by α-equivalence. We also equate nets obtained by "fusing" or "absorbing" wires:

$$[x, y] \mid [y, z] \equiv [x, z], \qquad\qquad \mu \mid [y, x] \equiv \mu\{x/y\} \quad \text{if } y \in \mathrm{fp}(\mu)$$

(note that $[x, y] \mid [y, x] \equiv [x, x]$ are legitimate nets, that is why nets and wires are *multisets*, not just sets).

An interaction net system is obtained by choosing an alphabet and fixing a set of *interaction rules* of the form

$$\alpha(x; \tilde{y}) \mid \beta(x; \tilde{z}) \quad \to \quad \nu\{\tilde{y}, \tilde{z}\},$$

where $\nu\{\tilde{y}, \tilde{z}\}$ is a net whose free ports are exactly $\tilde{y}, \tilde{z}$. An essential requirement is that there is *at most one* interaction rule for every unordered pair of symbols α, β in the alphabet. For instance, if we take $\mathbf{0}$, $\mathbf{s}$ and $+$ as symbols, of respective arities 0, 1 and 2, and if we fix the rules

$$\mathbf{0}(x) \mid +(x; y, z) \to \; [y, z], \qquad\qquad \mathbf{s}(x; u) \mid +(x; y, z) \to \; +(u; v, z) \mid \mathbf{s}(y; v),$$

we obtain a simple system for unary arithmetic with sum. Indeed, if we set $\underline{n}\{x\} = \mathbf{0}(v_1) \mid \mathbf{s}(v_2; v_1) \mid \cdots \mid \mathbf{s}(x; v_n)$, we invite the reader to check that $\underline{m}\{x\} \mid \underline{n}\{z\} \mid +(x; y, z) \to^* \underline{m+n}\{y\}$ in $m+1$ reduction steps. Although this particular example does not exhibit any parallelism (there is at most one reduction possible at each step), it is easy to imagine situations in which an arbitrary number of reductions may be fired at the same time.

We mentioned above that interaction nets have a simple and natural graphical representation, owing to their kinship with proof nets. Actually, in the existing literature nets are usually presented primarily in that way [10, 15, 16]. For instance, the rules for the above system for unary arithmetic would be defined graphically as in Fig. 1. In this paper, we stick to a textual representation, which has the advantage of being more concise and, we feel, more easily formalizable (although, arguably, much less visually appealing).

As a rewriting system, interaction nets are strongly confluent: this is because rewriting only acts on nets of the form $\alpha(x; \tilde{y}) \mid \beta(x; \tilde{z})$, called *active pairs*,

Fig. 1. The interaction rules for sum in unary arithmetic.

and these can never overlap (*i.e.*, there are no critical pairs) because x already appears twice and hence nowhere else. Strong confluence implies that Lafont's model is strictly deterministic. However, the parallelism of interaction nets suggests that, by endowing them with some form of non-determinism, it may be possible to obtain interesting models of concurrent computation.

The first to study such non-deterministic extensions was Vladimir Alexiev [1], who immediately realized that there are essentially three independent ways of altering Lafont's definition so as to inject non-determinism in the model:[3]

multirules: relaxing the requirement that there be at most one rule for every active pair;

multiwires: up to equivalence, an active pair has the form $\alpha(x, \tilde{s}) \mid \beta(y; \tilde{t}) \mid [x, y]$; if we allow wires connecting more than two ports, we obtain nets such as $\alpha(x; \tilde{s}) \mid \beta(y; \tilde{t}) \mid \gamma(z; \tilde{u}) \mid [x, y, z]$, in which active pairs overlap;

multiports: a further alternative is allowing agents to have more than one principal port, *i.e.*, more than one port on which they may interact with other cells. We thus obtain nets such as $\alpha(x, y; \tilde{s}) \mid \beta(x; \tilde{t}) \mid \gamma(y; \tilde{u})$, in which x and y are both principal for α, so one cell belongs to two active pairs.

Alexiev studied, to some extent, the inter-encodability of the various extensions, and exhibited an encoding of the replication-free π-calculus in the multiport variant, as proof that concurrent computation becomes possible in such extension of interaction nets.

In the ensuing years, other people independently defined or used similar non-deterministic variants of interaction nets, always in connection with concurrency: Ehrhard and Regnier's differential interaction nets [5] are in fact a special case of multirule interaction nets, in which Ehrhard and Laurent proposed an encoding of the π-calculus [4]; Beffara and Maurel's concurrent nets [3] use the multiwire extension, which is also mentioned by Yoshida in her work on concurrent combinators [22] and is implicit in the formulation of polarized proof nets used by Honda and Laurent to provide a correspondence with the asynchronous π-calculus [9]; and multiport interaction nets were shown by the second author to be able to encode the full π-calculus [17], improving Alexiev's result.

This leaves us with a natural question: are all these concurrent extensions of interaction nets equally expressive? Although, as mentioned above, encodings of

[3] Actually, Alexiev considered four extensions, but the fourth one has never been used in the literature.

the π-calculus were proposed for each one of these extensions, such encodings are so different in nature and their correctness is proved using such *ad hoc* arguments that, up to date, the relative expressiveness of each concurrent variant of interaction nets with respect to the others is far from clear.

The situation is further complicated by the absence, in concurrent interaction nets, of a notion of behavioral equivalence, an essential tool for comparing concurrent calculi. This is the ultimate reason why correctness proofs must resort to somewhat contrived arguments: in [4], correctness crucially depends on the definition of a labelled transition system on differential interaction nets which is quite *ad hoc* (if not highly questionable, see [18]); in [17], an operational correspondence between π-calculus reduction and interaction nets reduction is achieved through a notion of *readback* in interaction nets, which heavily depends on the encoding. Finally, although the authors of [9] do not need to address the problem because their operational correspondence is exact (*i.e.*, it is close to an "operational isomorphism"), the π-calculus they consider is asynchronous, while the other two encodings consider the synchronous one, and after Palamidessi's work [19] we know that the difference is not anodyne.

A comparison between the various non-deterministic extensions of interaction nets is attempted in the already mentioned work of Alexiev [1]. His conclusion is that the multiwire and multiport extensions are equivalent, whereas multirules are strictly less expressive. However, we feel that Alexiev's approach is not technically satisfactory: for the positive results, the question of defining a behavioral equivalence on interaction nets is not addressed and the correctness of the encodings is left unproven;[4] and the negative result is based on a severely constrained definition of translation (the nature principal/auxiliary of free ports must be preserved), which makes it less convincing than what one would hope. Finally, Alexiev never considers divergence, which is, as we will see, a key notion to capture the difference between multiwire and multiport nets.

In light of the above discussion, our starting point will be to propose a notion of behavioral equivalence for concurrent interaction nets, which is based in turn on giving a definition of "barb" in interaction nets. Our solution is to adopt a sort of "may testing" approach: we write $\mu{\downarrow}_x$ if there exists a net o such that $\mathrm{fp}(o) \cap \mathrm{fp}(\mu) = \{x\}$ and such that $\mu \mid o$ generates an "observable" computation. Since reduction rules in interaction net systems may be virtually *anything*, it is hopeless to define once and for all which computations are observable, regardless of the specific system. So we stipulate that observability comes with the definition of interaction net system itself: there is a non-empty set of "observable" interaction rules and an observable computation is a reduction sequence containing an observable reduction step (furthermore, we must require that, in $\mu \mid o$, such a sequence truly comes from the interaction of μ and o and is not already present in μ or o alone). In other words, our barbs are parametric in a choice of observable reduction rules.

[4] At p. 64 of [1], Alexiev states "[W]e don't prove formally the faithfulness of our translations, but we introduce them gradually and give comprehensive examples, so we hope that we have made their faithfulness believable".

Once barbs are given, barbed bisimulation and barbed congruence are defined in the standard way. Then, we proceed to introduce the notion of *translation* which will be the subject of our separation results. This is based on an almost straightforward reformulation, in interaction nets, of fairly standard properties which are asked of encodings between process algebras. We take as main reference Gorla's work [8], whose thorough analysis of the literature on encoding and separation results approaches exhaustiveness. Among other papers which are a guideline to our work we mention [20, 19].

In synthesis, the most important properties of our translations are the preservation of the degree of distribution, operational correspondence (completeness and correctness with respect to reductions, up to barbed congruence) and a bisimulation condition which excludes trivial encodings (such as those mapping every source net to the empty net).

Finally, our separation results technically take the the following form:

- there is a system of multiwire (or multiport) interaction net which cannot be translated into any interaction net system using only multirules (Theorem 1);
- there is a system of multiport interaction nets which cannot be translated into any interaction net system using only multirules and multiwires, *without introducing divergence* (Theorem 2).

The key to the first result is formalizing the fact that the multirule extension only provides interaction nets with "internal" non-determinism. For this, we introduce *must observability* $\mu\Downarrow_x$, which is defined by the fact that, for all μ' such that $\mu \to^* \mu'$, we have $\mu' \to^* \mu''\downarrow_x$. In other words, whatever happens inside μ, the port x will always be observable. Then, we verify that, in multirule systems, must observability may not be altered by interaction with contexts: if $\mu\Downarrow_x$ and ν does not contain x, then $(\mu \mid \nu)\Downarrow_x$. This is false in multiwire and multiport systems, and gives easily a separation argument.

The second result owes virtually everything to Palamidessi's idea for separating asynchrony from synchrony in the π-calculus [19]. Indeed, the proof is more or less a reformulation, in multiport interaction nets, of a simple leader election problem in a symmetric network, which we show to be translatable in multiwire systems only introducing divergence, because multiwires (and multirules) alone are not able to synchronously "break the symmetry".

Acknowledgments. This work was partially supported by ANR projects Panda (09-BLAN-0169-02) and Logoi (10-BLAN-0213-02). The second author wishes to dedicate this paper to the memory of Jamey Avon.

2 Concurrent Interaction Nets

Throughout the paper, we fix a denumerably infinite set of *ports*, ranged over by lowercase Latin letters. We write $\tilde{x}$ to denote a finite sequence of ports $x_1, \dots, x_n$ such that every port appears at most twice in the sequence; n is said to be the *length* of $\tilde{x}$. If ports appear at most once, we say that $\tilde{x}$ is *repetition-free*.

Definition 1 (Net). *An* alphabet *is a pair* $\Sigma = (|\Sigma|, \mathrm{deg})$*, where* $|\Sigma|$ *is a set and* $\mathrm{deg} : |\Sigma| \to \mathbb{N}$ *is the* degree function*.*

A cell*, or* agent*, on the alphabet* Σ *is an expression of the form* $\alpha(\tilde{x})$*, where* $\alpha \in |\Sigma|$ *and* $\tilde{x}$ *is of length* $\mathrm{deg}(\alpha)$*.*

A k-connector *is a multiset of cardinality* $k \in \mathbb{N}$ *of ports, containing at most two occurrences of every port, denoted by* $[\tilde{x}]$*. A 2-connector is called a* wire*; a* k*-connector with* $k = 1$ *or* $k \geq 3$ *is called a* multiwire*.*

A net *on an alphabet* Σ *is a finite multiset of connectors and agents on* Σ *in which every port appears at most twice. A net is* simply-wired *if it contains no multiwire.*

The set of free ports *of a net* μ*, denoted by* $\mathrm{fp}(\mu)$*, is the set of ports appearing exactly once in* μ*. The ports appearing twice in a net are called* bound*. We identify any two nets which may be obtained one from the other by an injective renaming of their bound ports (this is* α-equivalence*).*

We denote by $\mu\{y/x\}$ *the net* μ *in which the only free occurrence of* x *is replaced by* y*. The notation is extended to sequences (i.e.,* $\mu\{\tilde{y}/\tilde{x}\}$*) with the obvious meaning.*

Definition 2 (Juxtaposition). *Given two nets* μ, ν*, we denote by* $\mu \mid \nu$ *the net obtained by renaming (using* α*-equivalence) the bound ports of* μ *and* ν *so that the two nets have no bound name in common, and by taking then the standard multiset union.*

Note that, unlike usual process calculi, the symbol $\mid$ is not part of the syntax, it is an operation defined on nets. It is obviously commutative and has the empty net, denoted by 0, as neutral element. It is not associative in general; however, for $\mu \mid (\nu \mid \rho)$ and $(\mu \mid \nu) \mid \rho$ to be equal, it is enough that $\mathrm{fp}(\mu) \cap \mathrm{fp}(\nu) \cap \mathrm{fp}(\rho) = \emptyset$. More in general, if $\mu_1, \ldots, \mu_n$ are such that, for all pairwise distinct i, j, k, $\mathrm{fp}(\mu_i) \cap \mathrm{fp}(\mu_j) \cap \mathrm{fp}(\mu_k) = \emptyset$, then the expression $\mu_1 \mid \cdots \mid \mu_n$ is not ambiguous. Such a notation will always be used under this assumption in the sequel.

In the rest of the paper, by *congruence* on nets we mean an equivalence relation $\sim$ such that $\mu \sim \nu$ implies that for every net ρ, $\rho \mid \mu \sim \rho \mid \nu$.

Definition 3 (Structural congruence). Structural congruence*, denoted by* $\equiv$*, is the smallest congruence on nets satisfying the following:*

$$
\begin{array}{rrcll}
\text{0-connector:} & \mu \mid [] & \equiv & \mu & \\
\text{Fusion:} & [\tilde{x}, a] \mid [a, \tilde{y}] & \equiv & [\tilde{x}, \tilde{y}] & \\
\text{Wire:} & \mu \mid [a, x] & \equiv & \mu\{x/a\} & \quad \textit{if } a \in \mathrm{fp}(\mu)
\end{array}
$$

In the wire rule, we may further suppose that a appears in a cell (and not a connector) of μ, otherwise the rule is already subsumed by fusion.

It is sometimes useful to consider the "pure" structure of a net, abstracting from the specific names of its free ports. This is the reason behind the following notion.

Definition 4 (Mask). *We fix two infinite sequences of* reserved ports $(p_i)_{i \in \mathbb{N}}$ *and* $(q_i)_{i \in \mathbb{N}}$*. Any net on the alphabet* Σ *whose free ports are all reserved is called* *a* mask. *We suppose that no net other than a mask has reserved free ports. By* $\tilde{p}$ *and* $\tilde{q}$ *we will mean the sequences* $p_1, \ldots, p_m$ *and* $q_1, \ldots, q_n$*, resp., with* m *and* n *depending on the context.*

Quite obviously, every net μ whose free ports are in the repetition-free sequence $\tilde{x}$ may be seen as the "instantiation" of a mask μ_0, which is nothing but μ with its free ports suitably renamed: $\mu = \mu_0\{\tilde{x}/\tilde{p}\}$. The reason why we need a second sequence of reserved ports $(q_i)_{i\in\mathbb{N}}$ will be clarified shortly.

In what follows, we denote by $\widetilde{\mathcal{M}(\Sigma)}$ the set of finite repetition-free sequences of masks on Σ. We denote by $\|\xi\|$ the length of such a sequence ξ.

Definition 5 (Interaction scheme). *An* interaction scheme *on an alphabet* Σ *is a function* $\bowtie : |\Sigma| \times \mathbb{N} \times |\Sigma| \times \mathbb{N} \to \widetilde{\mathcal{M}(\Sigma)}$ *such that:*

1. *if* $\|\bowtie(\alpha, i, \beta, j)\| > 0$*, then* $1 \leq i \leq m = \deg\alpha$ *and* $1 \leq j \leq n = \deg\beta$*, and* $\alpha = \beta$ *implies* $i \neq j$*;*
2. *in that case, the* k*-th mask in the sequence* $\bowtie(\alpha, i, \beta, j)$ *is denoted by* $\alpha_i \overset{k}{\bowtie} \beta_j$ *and, for all* k*, its free ports are exactly* $p_1, \ldots, p_{i-1}$*,* p_{i+1}, p_m*,* $q_1, \ldots, q_{j-1}$*,* $q_{j+1}, \ldots, q_n$*;*
3. *furthermore, for all* (α, i, β, j)*,* $\|\bowtie(\beta, j, \alpha, i)\| = \|\bowtie(\alpha, i, \beta, j)\|$ *and, for all* $1 \leq k \leq \|\bowtie(\alpha, i, \beta, j)\|$*,* $\beta_j \overset{k}{\bowtie} \alpha_i = \alpha_i \overset{k}{\bowtie} \beta_j\{\tilde{q}/\tilde{p}, \tilde{p}/\tilde{q}\}$*.*

An interaction scheme defines rules to reduce active pairs. There may be several interaction rules for the same active pair, this is why $\bowtie(\alpha, i, \beta, j)$ is a list of nets, not just a net. Condition 2 says that the ports not partipating in the interaction are preserved by the rules. Condition 3 states that rules are symmetric when we swap symbols. Note that condition 1 stipulates that interaction rules are defined only between cells carrying different symbols or between different principal ports. This condition, which is present in the original definition of [10], was later relaxed by Lafont himself [12]. However, in this paper we adopt the more restrictive version, on the grounds that it is verified by all systems relevant to our work [5, 3, 9, 17].

Definition 6 (Interaction net system). *An* interaction net system *(INS) is a triple* $\mathcal{S} = (\Sigma_{\mathcal{S}}, \bowtie_{\mathcal{S}}, \mathcal{O}_{\mathcal{S}})$ *where* $\Sigma_{\mathcal{S}}$ *is an alphabet,* $\bowtie_{\mathcal{S}}$ *is an interaction scheme on* $\Sigma_{\mathcal{S}}$ *and* $\mathcal{O}_{\mathcal{S}} \subseteq |\Sigma_{\mathcal{S}}| \times \mathbb{N} \times |\Sigma_{\mathcal{S}}| \times \mathbb{N} \times \mathbb{N}$ *is non-empty and such that* $(\alpha, i, \beta, j, k) \in \mathcal{O}_{\mathcal{S}}$ *implies that* $\|\bowtie(\alpha, i, \beta, j)\| = l > 0$ *and* $1 \leq k \leq l$*, and that* $(\beta, j, \alpha, i, k) \in \mathcal{O}_{\mathcal{S}}$*. Subscripts are omitted when clear from the context.*

The set $\mathcal{O}_{\mathcal{S}}$ specifies the *observable rules* of $\mathcal{S}$: $(\alpha, i, \beta, j, k) \in \mathcal{O}_{\mathcal{S}}$ means that the k-th rule for the interaction between port i of an α cell and port j of a β cell is observable. The meaning of observable rules will be explained in Sect. 3.

Definition 7 (Reduction). *The* reduction relation $\to_{\mathcal{S}}$ *of an INS* $\mathcal{S}$ *is defined as follows:*

$$\frac{\alpha_i \overset{k}{\bowtie} \beta_j \ \textit{defined}}{\alpha(\tilde{x}) \mid \beta(\tilde{y}) \mid [x_i, y_j, z] \ \to_{\mathcal{S}} \ \alpha_i \overset{k}{\bowtie} \beta_j\{\tilde{x}/\tilde{p}, \tilde{y}/\tilde{q}\} \mid [z]} \ \text{INTERACTION}$$

$$\frac{\mu \to_{\mathcal{S}} \mu'}{\mu \mid \nu \to_{\mathcal{S}} \mu' \mid \nu}\ \text{CONTEXT} \qquad\qquad \frac{\mu \equiv \mu' \quad \mu' \to_{\mathcal{S}} \nu' \quad \nu' \equiv \nu}{\mu \to_{\mathcal{S}} \nu}\ \text{STRUCT}$$

We denote by $\to^*_{\mathcal{S}}$ *the reflexive-transitive closure of* $\to_{\mathcal{S}}$*. A net structurally congruent to the net on the left side of the* INTERACTION *rule is called an* (α_i, β_j)*-*active pair*. Clearly,* $\mu \to_{\mathcal{S}} \nu$ *only if some* (α_i, β_j)*-active pair is reduced, using the k-th rule for* (α, i, β, j)*. When we need to specify it, we write* $\mu \overset{\alpha_i\beta_j}{\longrightarrow}_k \nu$*.*

In an INS $\mathcal{S}$*, given* $\alpha \in |\Sigma_{\mathcal{S}}|$*, we say that the i-th port of* α *is* principal *if* $\|\bowtie_{\mathcal{S}}(\alpha, i, \beta, j)\| > 0$ *or* $\|\bowtie_{\mathcal{S}}(\beta, j, \alpha, i)\| > 0$ *for some* $\beta \in |\Sigma_{\mathcal{S}}|$*. Otherwise, it is called* auxiliary*.*

To improve readability, it is convenient to assume principal ports to be always the "leftmost" in the list of ports of a cell, and to use the notation $\alpha(x_1, \ldots, x_m; y_1, \ldots, y_n)$ for a cell whose symbol α is of degree $m + n$ and has m principal ports. If all ports are principal, the semicolon is omitted.

In practice, when defining an interaction net system it is convenient to specify the interaction scheme directly by giving rewriting rules of the form

$$\alpha(\tilde{x}) \mid \beta(\tilde{y}) \to \ \nu_1 + \cdots + \nu_l$$

where $\tilde{x}, \tilde{y}$ are repetition-free, $x_i = y_j = z$ for some i, j, and $\mathrm{fp}(\nu_k) = \{\tilde{x}, \tilde{y}\} \setminus \{z\}$. It is then intended that $\alpha_i \overset{k}{\bowtie} \beta_j$ is defined and equal to $\nu_k\{\tilde{p}/\tilde{x}, \tilde{q}/\tilde{y}\}$ (and this automatically defines also $\beta_j \overset{k}{\bowtie} \alpha_i$).

We conclude the section by introducing some terminology. An INS is:

- *multiport* if it has a symbol with more than one principal port; otherwise, it is *uniport*;
- *simply-wired* if all reduction rules introduce simply-wired nets (in that case, one usually restricts to simply-wired nets);

3 Barbs and Translations

In the following, we fix an arbitrary INS.

Definition 8 (Residue, interreduction). *Given an active pair* ϕ *of a net* μ *and a reduction* $\mu \to \ \mu'$ *reducing an active pair* ψ*, we have two possibilities: either* ϕ *and* ψ *share a cell (the extreme case being* $\phi = \psi$*), or they are disjoint. In the first case,* ϕ *has no residue in* μ'*; in the second case, the cells of* ϕ *are left untouched by the reduction, and* μ' *contains an active pair* ϕ' *which is "the same" as* ϕ*. This is its* residue *in* μ'*. The notion of residue is extended to reductions of arbitrary length in the obvious way.*

Let μ *be a net, and let* F *be a set of active pairs of* μ*. We say that a reduction* $\mu \to^* \mu'$ *is* F-legal *if it reduces no active pair of* F *nor any of their residues.*

Let μ, ν *be two nets, and let* F, G *be the set of* all *of their respective active pairs. An* interreduction *of* $\mu \mid \nu$ *is a reduction which is* $F \cup G$*-legal (juxtaposition may create active pairs not in* $F \cup G$*; this is why the definition is sensible).*

Definition 9 (Barbed bisimilarity). *Let $\mathcal{S}$ be an INS. We say that a reduction step $\mu \xrightarrow{\alpha_i\beta_j}_k \nu$ is* observable *if $(\alpha, i, \beta, j, k) \in \mathcal{O}_{\mathcal{S}}$.*

We write $\mu{\downarrow_x}$ if there exists a net o such that $\mathrm{fp}(\mu) \cap \mathrm{fp}(o) = \{x\}$ and an interreduction of $\mu \mid o$ containing an observable step. We write $\mu{\Downarrow_x}$ if $\mu \to^* \mu'{\downarrow_x}$ and we say that o is an observer *of x in μ.*

Let $\mathcal{S}, \mathcal{T}$ be two INSs. A (weak) barbed $(\mathcal{S}, \mathcal{T})$-bisimulation *is a binary relation $\mathcal{B} \subseteq \mathcal{S} \times \mathcal{T}$ on nets s.t. $\mathcal{B}(\mu, \nu)$ implies*

- *for every port x, $\mu{\downarrow_x}$ implies $\nu{\Downarrow_x}$ and $\nu{\downarrow_x}$ implies $\mu{\Downarrow_x}$;*
- *$\mu \to_{\mathcal{S}} \mu'$ implies that there exists ν' s.t. $\nu \to_{\mathcal{T}}^* \nu'$ and $\mathcal{B}(\mu', \nu')$;*
- *$\nu \to_{\mathcal{T}} \nu'$ implies that there exists μ' s.t. $\mu \to_{\mathcal{S}}^* \mu'$ and $\mathcal{B}(\mu', \nu')$.*

If there exists a barbed $(\mathcal{S}, \mathcal{T})$-bisimulation $\mathcal{B}$ such that $\mathcal{B}(\mu, \nu)$, we say that μ and ν are barbed bisimilar and write $\mu \;{}_{\mathcal{S}}\dot{\approx}_{\mathcal{T}}\; \nu$ (we drop the subscripts when the context is clear).

Barbed congruence *for $\mathcal{S}$, denoted by $\simeq^{\mathrm{c}}_{\mathcal{S}}$, is the greatest congruence contained in ${}_{\mathcal{S}}\dot{\approx}_{\mathcal{S}}$.*

The above definition of barb may be applied to standard name-passing calculi: there is only one reduction rule (i/o synchronization), which must be observable (by the definition, the set of observable rules is non-empty), and we thus obtain the usual barbs. The concept of interreduction is necessary to guarantee that the observable reduction step does not come from active pairs already present in μ or, worse, in the observer o.

In the following definition, by "net" we mean "net or mask".

Definition 10 (Translation). *Let $\mathcal{S}, \mathcal{T}$ be INSs. A* translation *from $\mathcal{S}$ to $\mathcal{T}$ is a map $[\![\cdot]\!]$ from nets of $\mathcal{S}$ to nets of $\mathcal{T}$ s.t., for all nets μ, μ' of $\mathcal{S}$:*

Homomorphism: *$[\![0]\!] = 0$ and $[\![\mu \mid \mu']\!] = [\![\mu]\!] \mid [\![\mu']\!]$;*

Port invariance: *for every mask μ_0 of $\mathcal{S}$, $\mathrm{fp}([\![\mu_0]\!]) = \mathrm{fp}(\mu_0)$, and if $\mu = \mu_0\{\tilde{x}/\tilde{p}\}$ (cf. observation after Definition 4), we have $[\![\mu]\!] = [\![\mu_0]\!]\{\tilde{x}/\tilde{p}\}$;*

Operational correspondence:
- *$\mu \to_{\mathcal{S}} \mu'$ implies $[\![\mu]\!] \to_{\mathcal{T}}^* \simeq^{\mathrm{c}}_{\mathcal{T}} [\![\mu']\!]$;*
- *$[\![\mu]\!] \to_{\mathcal{T}}^* \nu$ implies $\exists$ a net μ' of $\mathcal{S}$ s.t. $\mu \to_{\mathcal{S}}^* \mu'$ and $\nu \to_{\mathcal{T}}^* \simeq^{\mathrm{c}}_{\mathcal{T}} [\![\mu']\!]$;*

Bisimulation: *$\mu \;{}_{\mathcal{S}}\dot{\approx}_{\mathcal{T}}\; [\![\mu]\!]$.*

A translation does not introduce divergence *if, whenever $[\![\mu]\!]$ diverges, μ diverges.*

All three properties defining translations are more or less standard [8, 20]. The homomorphism condition guarantees that the degree of distribution is preserved by translations and is common in separation results [19]. Port invariance simply states that the interface of a net is preserved by a translation, and that the translation itself does not depend on the actual names of ports. Operational correspondence is a natural property to ask of an encoding, although we will not use it. On the contrary, the bisimulation condition will be essential. It corresponds to what Gorla [8] calls "success sensitiveness", in that it excludes trivial translations which would otherwise be validated by the other three conditions (such as an encoding mapping every net with free ports $x_1, \ldots, x_n$ to the net $[x_1], \ldots, [x_n]$). Furthermore, bisimulation (with the homomorphism property)

implies the adequacy and relative completeness of translations with respect to barbed congruence (of the respective systems):
adequacy: $[\![\mu]\!] \simeq^{c}_{\mathcal{T}} [\![\mu']\!]$ implies $\mu \simeq^{c}_{\mathcal{S}} \mu'$;
relative completeness: $\mu \simeq^{c}_{\mathcal{S}} \mu'$ implies $\forall$ net ρ of $\mathcal{S}$, $[\![\rho]\!] \mid [\![\mu]\!] \;{}_{\mathcal{T}}\dot{\approx}_{\mathcal{T}}\; [\![\rho]\!] \mid [\![\mu']\!]$.
This is a consequence of the (easy to verify) fact that, for any three INSs $\mathcal{S}, \mathcal{T}, \mathcal{U}$, $\mu \;{}_{\mathcal{S}}\dot{\approx}_{\mathcal{T}}\; \nu$ and $\nu \;{}_{\mathcal{T}}\dot{\approx}_{\mathcal{U}}\; \rho$ implies $\mu \;{}_{\mathcal{S}}\dot{\approx}_{\mathcal{U}}\; \rho$.

4 Multirules Alone Do Not Give Concurrency

In the following, we fix an arbitrary INS $\mathcal{S}$.

Definition 11 (Must observability). *A port x is said to be* must-observable *in the net μ if, for all μ' s.t. $\mu \to^* \mu'$, we have $\mu' \Downarrow_x$. In that case, we write $\mu \Downarrow\!\!\Downarrow_x$.*

Observe that, by definition, must observability is preserved by reduction.

Lemma 1. *Let x be a port and let $\mu \equiv \mu' \mid \alpha(y; \tilde{z})$ be a net of $\mathcal{S}$, with x different from y and all of the ports in $\tilde{z}$, and $y \notin \mathrm{fp}(\mu')$. Then, $\mu \Downarrow\!\!\Downarrow_x$ iff $\mu' \Downarrow\!\!\Downarrow_x$.*

Proof. The cell $\alpha(y; \tilde{z})$ may react only on y, but y is free in μ, so the cell does not participate in any reduction of μ. □

For technical reasons, we introduce the following restricted notion of barbed bisimulation:

Definition 12 (x-bisimulation). *Let x be a port. An x-*bisimulation *is a binary relation $\mathcal{B}$ on nets of $\mathcal{S}$ such that, whenever $\mathcal{B}(\mu, \nu)$, $\mu \downarrow_x$ implies $\nu \Downarrow_x$ and $\nu \downarrow_x$ implies $\mu \Downarrow_x$, plus the usual reduction properties required by barbed bisimulations (last two points of Definition 9).*

In other words, an x-bisimulation is a usual barbed bisimulation in which we content ourselves with simulating barbs on x only.

Lemma 2. *Let $\mathcal{B}$ be an x-bisimulation, and let $\mathcal{B}(\mu, \nu)$. Then, $\mu \Downarrow\!\!\Downarrow_x$ iff $\nu \Downarrow\!\!\Downarrow_x$.*

Proof. Immediate. □

Lemma 3. *Suppose that $\mathcal{S}$ is uniport and simply-wired, and let μ be a simply-wired net of $\mathcal{S}$ such that $\mu \Downarrow\!\!\Downarrow_x$. Then, for every simply-wired net ν such that $x \notin \mathrm{fp}(\nu)$, $(\mu \mid \nu) \Downarrow\!\!\Downarrow_x$.*

Proof. By definition, $\mathcal{O}_{\mathcal{S}} \neq \emptyset$, so let $(\alpha, 1, \alpha', 1, k) \in \mathcal{O}_{\mathcal{S}}$. Let $\tilde{y}$ be a repetition-free sequence not containing x, of length equal to the number of auxiliary ports of α, and consider the relation

$$\mathcal{B} = \{(\mu \mid \nu, \alpha(x; \tilde{y}))\ ;\ \mu, \nu \text{ simply-wired},\ \mu \Downarrow\!\!\Downarrow_x,\ x \notin \mathrm{fp}(\nu)\}.$$

We claim that $\mathcal{B}$ is an x-bisimulation. Let $(\mu \mid \nu, \alpha(x; \tilde{y})) \in \mathcal{B}$. First of all, $\mu \Downarrow\!\!\Downarrow_x$ implies $\mu \Downarrow_x$ which implies $(\mu \mid \nu) \Downarrow_x$, and by hypothesis $\alpha(x; \tilde{y}) \downarrow_x$, so the first two properties are met. Since $\alpha(x;\tilde{y})$ does not reduce, it is enough to show how $\alpha(x;\tilde{y})$ simulates a reduction $\mu \mid \nu \to \rho$. Such a reduction necessarily comes from an active pair ϕ. If ϕ is entirely contained in μ or ν, the definition of $\mathcal{B}$ allows us to conclude immediately. So we suppose that ϕ is an active pair created by the juxtaposition of μ and ν, *i.e.*, we may assume that

$$\mu \equiv \mu' \mid \beta(z;\tilde{t}),$$
$$\nu \equiv \gamma(z;\tilde{s}) \mid \nu',$$

with z free both in μ and ν, because both nets are simply-wired. Then, if ρ' is either $\beta_1 \overset{k}{\bowtie} \gamma_1\{\tilde{t}/\tilde{p},\tilde{u}/\tilde{q}\}$ or $\gamma_1 \overset{k}{\bowtie} \beta_1\{\tilde{u}/\tilde{p},\tilde{t}/\tilde{q}\}$ (for some irrelevant k), we have $\rho = \mu' \mid \rho' \mid \nu'$. But by Lemma 1, $\mu'\Downarrow_x$, so $(\rho, \alpha(x;\tilde{y})) \in \mathcal{B}$ by definition of $\mathcal{B}$.

Now, obviously $\alpha(x;\tilde{y})\Downarrow_x$ (as already observed, we have $\alpha(x;\tilde{y})\downarrow_x$ and the net does not reduce), so we may conclude by Lemma 2. □

Lemma 3 is false in presence of multiwires or multiports. For instance, consider an INS in which there are two symbols α, β, of degree 1 and 2, respectively, with the following interaction rule (which is observable, since it is the only one):

$$\alpha(x) \mid \beta(x;y) \to \alpha(y).$$

If we set $\mu = \alpha(x) \mid [x,y,z]$, we obviously have $\mu\Downarrow_y$ and $\mu\Downarrow_z$. However, for example, although still observable, z is no longer must-observable in $\mu \mid \beta(y;s)$, because $\mu \mid \beta(y;s) \to \alpha(s) \mid [z]$, in which there is no way to observe z. Similar examples may be built with multiports.

Theorem 1. *There exists an INS $\mathcal{S}$ which cannot be translated into any simply-wired, uniport INS $\mathcal{T}$ using only simply-wired nets.*

Proof. Take as $\mathcal{S}$ the system defined above, in which we allow nets containing multiwires, and suppose there exists a translation $[\![\cdot]\!]$ into a simply-wired, uniport INS $\mathcal{T}$ whose image consists of simply-wired nets only. Let $\mu = \alpha(x) \mid [x,y,z]$. Since $\mu \dot{\approx} [\![\mu]\!]$, we must have $[\![\mu]\!]\Downarrow_z$. Consider now the net $\rho = \mu \mid \beta(y;s)$. By the homomorphism property, $[\![\rho]\!] = [\![\mu]\!] \mid [\![\beta(y;s)]\!]$. By port preservation, $x \notin \mathrm{fp}([\![\beta(y;s)]\!])$, so we may apply Lemma 3 (all nets in the image of the translation are simply wired) and infer that $[\![\rho]\!]\Downarrow_z$. But we saw above that we do not have $\rho\Downarrow_z$, contradicting the fact that $\rho \dot{\approx} [\![\rho]\!]$. □

As already mentioned, although the system $\mathcal{S}$ used in the proof is uniport and uses multiwires, there is no difficulty in finding a simply-wired but multiport system $\mathcal{S}'$ for which Theorem 1 holds (with basically the same proof).

5 Comparing Multiwire and Multiport Concurrency

Definition 13 (Symmetric net). *A net μ is* strictly symmetric *if there exists a net ν whose free ports contain (but do not necessarily coincide with) $\tilde{s},\tilde{t},\tilde{u}$ such that*

$$\mu = \nu\{\tilde{a}/\tilde{s},\tilde{a}'/\tilde{t},\tilde{x}/\tilde{u}\} \mid \nu\{\tilde{a}'/\tilde{s},\tilde{a}/\tilde{t},\tilde{x}'/\tilde{u}\}.$$

In that case, the free ports of μ *are* $\tilde{x}, \tilde{x}'$*, and the pairs of ports* x_i, x'_i *are said to be* exchanged *by the symmetry. We say that* μ *is* symmetric *if* $\mu \equiv \mu_0$ *with* μ_0 *strictly symmetric.*

Symmetric nets enjoy the following three fundamental properties: they are preserved by translations (if they are strict), their barbs always "come in pairs" and, if we are in a uniport system, there is no way of irreversibly breaking the symmetry in just one reduction step.

Lemma 4. *If* μ *is a strictly symmetric net and* $[\![\cdot]\!]$ *is a translation,* $[\![\mu]\!]$ *is strictly symmetric too.*

Proof. An immediate consequence of the homomorphism and port invariance properties of translations. □

Lemma 5. *Let* μ *be a symmetric net and let* $x, x' \in \mathrm{fp}(\mu)$ *be exchanged by the symmetry. Then:*

- $\mu{\downarrow_x}$ *iff* $\mu{\downarrow_{x'}}$*;*
- $\mu{\Downarrow_x}$ *iff* $\mu{\Downarrow_{x'}}$*.*

Proof. If $\tilde{y}, \tilde{y}'$ are the free ports of μ, with y_i, y'_i exchanged by the symmetry, then $\mu = \mu\{\tilde{y}'/\tilde{y}, \tilde{y}/\tilde{y}'\}$. The result then follows immediately. □

Lemma 6. *Let* μ *be a symmetric net in a uniport INS and let* $\mu \to \nu$*. Then, there exists a symmetric net* ν' *s.t.* $\nu \to \nu'$*.*

Proof. Let $\mu \equiv \rho \mid \rho'$, with ρ, ρ' instances of the same net ρ_0 as in Definition 13. If the active pair ϕ reduced to obtain ν is entirely in one of the two symmetric components of μ, it has a counterpart ϕ' in the other component, which obviously has a residue in ν (*cf.* Definition 8), by reducing which, in the same way as ϕ, we obtain a symmetric net ν'. Otherwise, the active pair ϕ is created by the juxtaposition of the two copies of ρ_0, so we have $\rho = \rho_1 \mid \alpha(a; \tilde{b})$, $\rho' = \beta(a; \tilde{b}') \mid \rho'_1$ and ϕ is composed by the α and β cells. But $\alpha \neq \beta$ (condition 1 of Definition 5) so we must actually have $\rho_0 = \pi \mid \alpha(s; \tilde{u}) \mid \beta(t; \tilde{v})$, *i.e.*, there is a β cell in ρ_1 and an α cell in ρ'_1. By symmetry, these form an active pair ϕ' in μ which is of the same nature as ϕ. Again, ϕ' has a residue in ν by reducing which (in the same way as ϕ) we obtain a symmetric net ν'. □

Lemma 6 is false in multiport systems. Consider the INS $\mathcal{S}$ defined as follows: its alphabet consists of two symbols α, with two principal ports and one auxiliary port, and υ, with one principal and one auxiliary port; the reduction rules are

$$\alpha(a, s; x) \mid \alpha(t, a; y) \to \upsilon(x; s) \mid [t, y],$$

and any rule for $\alpha(a, t; x) \mid \upsilon(a; y)$, which is observable. The idea is that when two α cells "meet", one on its first and the other on its second principal port, the one which interacts on the first principal port "wins". Victory is represented by the fact that its auxiliary port becomes the principal port of a υ cell, which is observable by virtue of the (otherwise irrelevant) observable rule for $(\alpha, 1, \upsilon, 1)$.

Let now

$$\mho = \alpha(a, b; x) \mid \alpha(b, a; y),$$

which is obviously strictly symmetric. We have $\mho \to\ \upsilon(x; y) = \mu_x$ and $\mho \to \upsilon(y; x) = \mu_y$, both of μ_x and μ_y do not reduce further and neither of them is symmetric. In fact, they are such that $\mu_x{\downarrow_x}$ but $\mu_x{\not\downarrow_y}$, whereas $\mu_y{\downarrow_y}$ but $\mu_y{\not\downarrow_x}$.

Theorem 2. *There exists an INS $\mathcal{S}$ which cannot be translated into any uniport INS without introducing divergence.*

Proof. We take as $\mathcal{S}$ the multiport system just introduced above and we consider the net we denoted by $\mho$. Let $[\![\cdot]\!]$ be a translation of $\mathcal{S}$ into a uniport INS and let $\nu_0 = [\![\mho]\!]$. By Lemma 4, ν_0 is symmetric and x, y are exchanged by its symmetry. By the bisimulation property, we know that there exists a barbed bisimulation $\mathcal{B}$ such that $\mathcal{B}(\mho, \nu_0)$. Since $\mho \to\ \mu_x$, we must have $\nu_0 \to^* \nu_x$ such that $\mathcal{B}(\mu_x, \nu_x)$. Since $\mu_x{\downarrow_x}$ but $\mu_x{\not\downarrow_y}$, by Lemma 5 we must have $\nu_x \neq \nu_0$, which means that at least one reduction step is possible from ν_0. Then, we may apply Lemma 6 and infer that $\nu_0 \to^* \nu_1$ in at least one reduction step, with ν_1 symmetric. But this implies that $\mho \to^* \mu_1$ such that $\mathcal{B}(\mu_1, \nu_1)$. Now, μ_1 can only be one of μ_x, μ_y or $\mho$ itself, but the symmetry of ν_1 and Lemma 5 rule out the first two cases, hence $\mathcal{B}(\mho, \nu_1)$.

The reader is invited to check that, in the above reasoning, we deduced $\mathcal{B}(\mho, \nu_1)$ starting from $\mathcal{B}(\mho, \nu_0)$ using only the fact that ν_0 is symmetric and that its two free ports are x, y (and must therefore be exchanged by its symmetry). These properties still hold for ν_1, so we may apply the reasoning again and again, obtaining a reduction sequence $[\![\mho]\!] = \nu_0 \to^* \nu_1 \to^* \nu_2 \to^* \cdots$, in which every reduction $\nu_i \to^* \nu_{i+1}$ is of length at least 1, so $[\![\mho]\!]$ diverges. □

6 Discussion

Significance. A potentially controversial point of our definition of barb is its parametricity, which makes barbed congruence somewhat arbitrary. A possible answer is the following: there is always a default choice, which consists in deeming *every* rule observable. Concretely, a default barb $\mu{\downarrow_x}$ is equivalent to the fact that x is a free principal port in μ, *i.e.*, $\mu = \alpha(\tilde{x}; \tilde{y}) \mid \mu'$ with $x = x_i$ for some i and α.

Default barbed congruence is analogous to usual barbed congruence in standard process calculi, including the solos calculus [14], of which interaction nets are strongly reminiscent.[5] The possibility of using smaller sets of observable rules should only be considered in encodings: if an INS $\mathcal{S}$, with its default barbed congruence, is to be encoded in an INS $\mathcal{T}$, it may be reasonable to consider instead $(\Sigma_{\mathcal{T}}, \bowtie_{\mathcal{T}}, \mathcal{O})$, where we exclude from $\mathcal{O}$ the "administrative" rules of $\mathcal{T}$, thus

[5] In fact, by considering two families of symbols ι_n, o_n with the rules $\iota_n(x; \tilde{y}) \mid o_n(x; \tilde{z}) \to\ [y_1, z_1] \mid \cdots \mid [y_n, z_n]$, one basically obtains the replication-free solos calculus with explicit fusions [21], with names represented by multiwires.

weakening barbed congruence (in the extreme case $\mathcal{O} = \emptyset$, which is not allowed by our definition, barbed congruence would equate everything).

It is also interesting to consider default barbed congruence in Lafont interaction nets systems, *i.e.*, the strictly deterministic kind, for which definitions of observational equivalences already exist [6]. In this setting, we are able to prove that, if we ignore the notion of "constructor symbol" used by Fernández and Mackie (which has no counterpart in our definitions), default barbed congruence coincides with their observational equivalence. So, at least in the simple deterministic case, our definitions fall back on something already known to be meaningful.

As far as our notion of translation is concerned, it is based on properties which are mostly agreed upon in the literature [8, 20]. The only other existing notion of translation for interaction nets is the one mentioned above, formulated by Lafont for his deterministic systems [10]. It is possible to show that, if we consider default barbed congruence, a Lafont's translation induces a translation in our sense. Conversely, thanks to determinism, our definition does not differ from Lafont's one in an essential way, although it is more permissive.

Turning to our separation results, as they are technically formulated, Theorem 1 and Theorem 2 may be easily criticized: even if we agree on the reasonability of our notion of translation, the sole existence of an untranslatable system may not be enough to convincingly separate two extensions; it all depends on the relevance of such a system.

We believe that the relevance of the untranslatable systems is given by Lemma 3, for Theorem 1, and Lemma 6, for Theorem 2. In both cases, there is one "limiting" property which always holds in one extension of interaction nets but fails in the "more expressive" ones. In the first case, the limitation is so severe that we are led to conclude that multirules alone cannot express concurrency: indeed, Lemma 3 is false in any standard process calculus. On the other hand, Lemma 6 shows the same limitation pointed out by Palamidessi for the asynchronous π-calculus: in absence of multiports, interaction nets are unable to make certain irreversible choices in just one step (*i.e.*, synchronously). Instead, such choices must always involve a reversible "pre-commitment" phase. Only once such a phase is successfully concluded may the choice be irreversibly committed. Theorem 2 shows this for an "electoral system" with only two nodes, but the argument scales up to arbitrarily large "leader-election" nets, as in [19].

Concluding remarks. We observe that our first separation result casts doubts on the value of the encoding of the π-calculus in differential interaction nets (which are a multirule INS) given in [4]. As already pinpointed by the second author [18], that encoding supposes a labelled semantics of differential interaction nets which is not "realistic" in terms of concurrency.

We are also left with an interesting open question concerning the relaxation of condition 1 of Definition 5, allowing "self-interaction", as considered by Lafont [12]. We know that Lemma 6 fails in presence of such a relaxation. In process calculi, this would correspond to introducing "neutral" prefixes, neither input nor output, which may synchronously interact with each other. Palamidessi's symmetry argument then does not apply straightforwardly and "neutral" synchronization might be as expressive as input/output synchronization.

References

1. Alexiev, V.: Non-deterministic Interaction Nets. Ph.D. Thesis, University of Alberta (1999)
2. Asperti, A., Guerrini, S.: The Optimal Implementation of Functional Programming Languages. Cambridge University Press (1998)
3. Beffara, E., Maurel, F.: Concurrent nets: A study of prefixing in process calculi. Theoretical Computer Science 356(3), 356–373 (2006)
4. Ehrhard, T., Laurent, O.: Interpreting a Finitary Pi-Calculus in Differential Interaction Nets. Information and Computation 208(6), 606–633 (2010)
5. Ehrhard, T., Regnier, L.: Differential Interaction Nets. Theoretical Computer Science 364(2), 166–195 (2006)
6. Fernández, M., Mackie, I.: Operational Equivalence for Interaction Nets. Theoretical Computer Science 297(1–3), 157–181 (2003)
7. Gonthier, G., Abadi, M., Lévy, J.J.: The geometry of optimal lambda reduction. In: Sethi, R. (ed.) Proceedings of POPL. pp. 15–26. ACM Press (1992)
8. Gorla, D.: Towards a unified approach to encodability and separation results for process calculi. Information and Computation 208(9), 1031–1053 (2010)
9. Honda, K., Laurent, O.: An exact correspondence between a typed pi-calculus and polarised proof-nets. Theor. Comput. Sci 411(22-24), 2223–2238 (2010)
10. Lafont, Y.: Interaction nets. In: Allen, F.E. (ed.) Proceedings of POPL. pp. 95–108. ACM Press (1990)
11. Lafont, Y.: From proof nets to interaction nets. In: Girard, J.Y., Lafont, Y., Regnier, L. (eds.) Advances in Linear Logic, pp. 225–247. Cambridge University Press (1995)
12. Lafont, Y.: Interaction Combinators. Information and Computation 137(1), 69–101 (1997)
13. Lamping, J.: An algorithm for optimal lambda calculus reduction. In: Allen, F.E. (ed.) Proceedings of POPL. pp. 16–30. ACM Press (1990)
14. Laneve, C., Victor, B.: Solos In Concert. Mathematical Structures in Computer Science 13(5), 657–683 (2003)
15. Mackie, I.: An Interaction Net Implementation of Additive and Multiplicative Structures. Journal of Logic and Computation 15(2), 219–237 (2005)
16. Mackie, I.: An interaction net implementation of closed reduction. In: Scholz, S.B., Chitil, O. (eds.) Proceedings of IFL. LNCS, vol. 5836, pp. 43–59. Springer (2008)
17. Mazza, D.: Multiport Interaction Nets and Concurrency. In: Abadi, M., de Alfaro, L. (eds.) Proceedings of CONCUR 2005. pp. 21–35. LNCS, Springer (2005)
18. Mazza, D.: The True Concurrency of Differential Interaction Nets. Mathematical Structures in Computer Science (2013), accepted for publication. To appear.
19. Palamidessi, C.: Comparing the expressive power of the synchronous and asynchronous pi-calculi. Math. Structures Comput. Sci. 13(5), 685–719 (2003)
20. Parrow, J.: Expressiveness of Process Algebras. Electronic Notes in Theoretical Computer Science 209, 173–186 (2008)
21. Wischik, L., Gardner, P.: Explicit fusions. Theoretical Computer Science 340(3), 606–630 (2005)
22. Yoshida, N.: Graph notation for concurrent combinators. In: Ito, T., Yonezawa, A. (eds.) Proceedings of TPPP. LNCS, vol. 907, pp. 393–412. Springer (1994)